Computer Communications 89–90 (2016) 154–164

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

Practical access control for sensor networks in the context of the Internet of Things☆

Fagen Li*, Yanan Han, Chunhua Jin

School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China

ARTICLE INFO

Article history:
Available online 15 March 2016

Keywords:
Internet of Things
Security
Signcryption
Certificateless cryptography
Identity-based cryptography

ABSTRACT

Wireless sensor network (WSN) plays an important role in military sensing and tracking, target tracking, and environment monitoring. To query of the network to get useful information from anywhere and anytime, we need to integrate the WSN into the Internet as part of the Internet of Things (IoT). In this case, it is an important task to design an access control scheme that can authorize, authenticate and revoke a user to access the WSN. In this paper, we propose a heterogeneous signcryption scheme to control the access behavior of the users. We give the formal security proof of our scheme in the random oracle model. An important characteristic of our scheme is to allow a user in a certificateless cryptography (CLC) environment to send a message to a sensor node in an identity-based cryptography (IBC) environment. We give an access control scheme for the WSN in the context of the IoT using the proposed signcryption scheme. As compared with existing two access control schemes using signcryption, the computational cost of sensors in our scheme is reduced by about 22% and 53%, respectively and the energy consumption of sensors in our scheme is reduced by about 33% and 54%, respectively.

1. Introduction

Wireless sensor networks (WSNs) are ad hoc networks which usually are composed of a large number of tiny sensor nodes with the capabilities of sensing, computation and communication [1]. WSNs have important application value in military sensing and tracking, target tracking, environment monitoring, and so on. For example, we can deploy a WSN to monitor the efficiency of each industrial equipment by measuring vibration, temperature, pressure, power quality, and so on. If a factory personnel find a potential problem by collecting the data from the WSN, he or she may repair or replace the equipment before the efficiency of the equipment drops or the equipment fails entirely. As compared with the traditional wired industrial monitoring system, industrial WSNs have lower cost for development and maintenance and higher flexibility and intelligent process capability [2,3]. While the WSNs provide a great flexibility for establishing communications, it also brings a lot of technical challenges. In [2], Gungor and Hancke gave eight technical challenges for the WSNs. The fifth challenge is the security due to all the characteristics of these networks, such as open nature of wireless communication, dynamically changing topology, and the limited capabilities of sensor nodes in terms of processing power, storage, bandwidth, and energy. The eighth challenge is the integration with the Internet. To query of the WSNs to get useful information from anywhere and anytime, we need to integrate the WSNs into the Internet as part of the Internet of Things (IoT). We have three methods to achieve this integration, front-end proxy solution, gateway solution and TCP/IP overlay solution [4]. In the front-end proxy solution, the sensor nodes cannot communicate with the Internet hosts directly. The base station acts as an interface between the WSNs and the Internet and parses all incoming and outgoing information. That is, the users issue data queries to sensor nodes through the base station and the base station forwards the corresponding results to the users. The weakness of this solution is that the base station may become the bottleneck and the single point of failure. In both gateway solution and TCP/IP overlay solution, the sensor nodes can directly communicate with the Internet hosts. In the gateway solution, the base station plays the role of an application layer gateway that translates the lower layer protocols from both networks. In the TCP/IP overlay solution,

☆ This work is supported by the National Natural Science Foundation of China (grant nos. 61073176, 61272525, 61302161 and 61462048) and the Fundamental Research Funds for the Central Universities (grant no. ZYGX2013J069).

* Corresponding author.
E-mail address: fagenli@uestc.edu.cn (F. Li).

http://dx.doi.org/10.1016/j.comcom.2016.03.007

sensor nodes use TCP/IP to communicate with other nodes. The base station plays the role of a router that forwards the packets from and to the sensor nodes.

In order to prevent abuse of the data collected by the WSNs, only authorized users are allowed to access the WSNs. However, it is not an easy thing to design an access control scheme for the WSNs in the context of the IoT since the resource of the sensor nodes is very limited.

1.1. Related work

In 2009, Le et al. [5] proposed an energy-efficient access control scheme for the WSNs using elliptic curve cryptography. The advantage of elliptic curve cryptography is that it can afford comparable security level to the other public key cryptography such as RSA [6] with smaller key size. For example, to obtain the 80-bit security level, the modulus size of RSA should be 1024 bits but the key size of elliptic curve cryptography only needs 160 bits. In 2011, He et al. [7] proposed a privacy-preserving access control scheme for the WSNs using ring signature technique [8,9]. The ring signature allows a signer to anonymously sign a message on behalf of a set of users including itself. It protects the privacy of the signer since the verifier knows that the message comes from a member of a ring, but does not know exactly who the signer is. In 2012, Zhang et al. [10] gave a new solution for privacy-preserving access control scheme for the WSNs using blind signature technique. Yu et al. [11] gave a fine-grained data access control scheme for the WSNs using attribute-based encryption [12]. Recently, Yu et al. [13] (hereafter called YHZXZ) and Ma et al. [14] (hereafter called MXH) used signcryption [15] approach to design the access control for the WSNs. The use of signcryption is very novel and efficient for the WSNs application because it simultaneously authenticates the users and protects the query messages with a low cost. Signcryption is a cryptographic primitive that performs both the functions of digital signature and public key encryption in a logical single step, with a cost significantly lower than that required by the traditional signature-then-encryption or encryption-then-signature methods. That is, signcryption can simultaneously accomplishes confidentiality, integrity, authentication and non-repudiation with a lower cost. However, these schemes [13,14] are based on the traditional public key infrastructure (PKI). In the PKI, each user has a private key and a corresponding public key. In order to ensure the authenticity of the public key, a certificate authority (CA) needs to issue a digital certificate that affords an unforgeable and trusted link between the public key and the identity of a user by the digital signature of the CA. The main difficulty of the traditional PKI is the certificates management, including distribution, storage and revocation. Furthermore, each user should verify the validity of a certificate before using the public key described in this certificate. If the certificate is valid, the public key is believable and the user can use it. Otherwise, the user cannot use the public key in any cryptographic operation. For the access control for WSNs in the context of the IoT, it is a heavy burden for the sensor nodes to verify the validity of the public key certificates. To reduce the burden of the sensor nodes, identity-based cryptography (IBC) [16] was proposed to design the security schemes for the WSNs [17–20]. As compared with the traditional PKI, the main advantage of the IBC is the elimination of public key certificates. In the IBC, a user's public key is computed from its identity information, such as telephone numbers, IP addresses and email addresses. There exist a trusted third party called private key generator (PKG) who is in charge of the generation of private keys for all users. Authenticity of a public key is explicitly verified without a certificate. So the IBC is a good choice for design the security for the WSNs. However, the lightweight IBC has the key escrow problem since the PKG knows all users' private keys [16]. IBC is only suitable for small networks, such as the WSNs, and is not suitable for large-scale networks, such as the Internet. For design an access control scheme for the WSNs in the context of the IoT, a possible solution is that the WSNs part uses the IBC and the Internet part uses the PKI. There exist such signcryption schemes [21,22]. However, in these solutions, when an Internet user issues a query to the WSNs, the sensor nodes need to check if the Internet user has been authorized. The sensor nodes still need to verify the public key certificate since the Internet user belongs to the PKI.

1.2. Motivation and contribution

The motivation of the paper is to design a practical access control scheme without certificates for WSNs in the context of the IoT. Only authorized Internet users can access the WSNs and the query messages are protected. The protection of the query messages is very important for preserving the privacy of the users [14]. Different to [13] and [14], our solution uses a novel heterogeneous signcryption (HSC) in which the senders belong to the certificateless cryptography (CLC) environment [23] and the receivers belong to the IBC environment. The CLC does not require the use of certificates and yet does not has the built-in key escrow problem of IBC. The CLC still needs a trusted third party called the key generating center (KGC) who is responsible for generating a partial private key using a user's identity and a master key. The user then combines the partial private key with some secret value to generate a full private key which is unknown to the KGC. We show that the novel heterogeneous signcryption scheme has the indistinguishability against adaptive chosen ciphertext attacks (IND-CCA2) under the gap bilinear Diffie–Hellman (GBDH) problem and existential unforgeability against adaptive chosen messages attacks (EUF-CMA) under the gap Diffie–Hellman (GDH) and computational Diffie–Hellman (CDH) problems in the random oracle model. An important characteristic of our scheme is heterogeneous. That is, senders and receivers belong to two different cryptographic environments. It allows a sender in the CLC environment to transmit a message to a receiver in the IBC environment. In addition, our scheme has the ciphertext authenticity that allows we shift the computational cost of the sensor nodes to the gateway. We give an access control scheme for the WSNs in the context of the IoT using the novel signcryption scheme. As compared with existing two access control schemes using signcryption, the computational cost of the sensor node in our scheme is reduced by about 22% and 53%, respectively and the energy consumption of the sensor node in our scheme is reduced by about 33% and 54%, respectively.

1.3. Organization

The rest of the paper is arranged as follows. The network model, security requirements and bilinear pairings are introduced in Section 2. The novel heterogeneous signcryption scheme is proposed in Section 3. We give the access control scheme for the WSNs in the context of the IoT in Section 4. Finally, the conclusions are given in Section 5.

2. Preliminaries

In this section, we give the network model, security requirements and bilinear pairings.

2.1. Network model

Fig. 1 shows the overview of the network model. The model consists of four kinds of entities, a service provider (SP), the sensor nodes, a gateway and the Internet hosts (users). The SP deploys

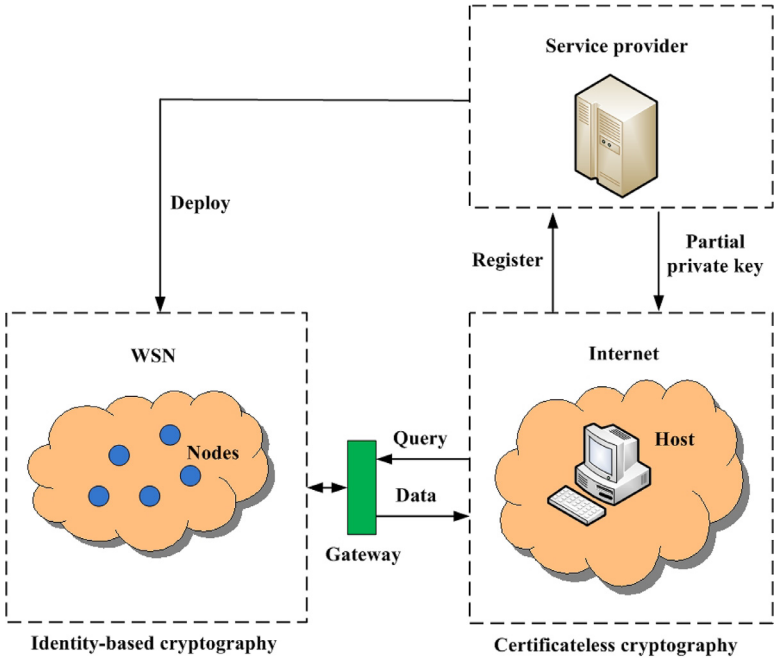

Fig. 1. Network model.

a WSN that collects the monitoring data. The users who want to access the WSN should be authorized by the SP. Anyone without authorization cannot access the WSN. The SP takes charge of the registration for sensor nodes and users and generates the private keys for sensor nodes and partial private keys for users. That is, the SP plays the role of PKG in the IBC environment and the KGC in the CLC environment. The sensor nodes have limited computational power and storage resource while the gateway has higher processing and storage capability. We assume that the SP is always trusted and can never be compromised and the gateway is honest and curious. When a user hope to access the data of the WSN, it sends a query message to a sensor node. The gateway first checks if the user has been authorized to access the data. If yes, the gateway forwards the query to the sensor node and the node transmits collected data to the user in a secure way. Otherwise, the gateway rejects the query request. Our scheme proposed in this paper can be used in both gateway solution and TCP/IP overlay solution.

2.2. Security requirements

The communication between the Internet hosts and sensor nodes should satisfy confidentiality, integrity, authentication, and non-repudiation. Confidentiality is keeping query messages secret from the others except the Internet hosts and sensor nodes. Even the gateway cannot know the contents of the message. Integrity is ensuring that the query messages from the Internet hosts have not been altered by unauthorized entities. Authentication is the assurance that only the authorized Internet hosts can access the WSN. Non-repudiation is preventing the denial of previous queries issued by the Internet hosts. That is, if an Internet host has sent a query message to a sensor node, it cannot deny its action. In addition, we hope that this communication also satisfies ciphertext authenticity and insider security. The ciphertext authenticity means that a third party can verify the validity of a ciphertext without knowing the query message. The insider security has two aspects. The first aspect is the insider security for confidentiality and the other aspect is the insider security for unforgeability. The insider security for confidentiality guarantees the forward security of signcryption, i.e. confidentiality is kept in case the host's private key is compromised. The insider security for unforgeability means that an adversary cannot forge a ciphertext with the sensor node's private key.

2.3. Bilinear pairings

Let G_1 and G_2 be two cyclic groups with same prime order p. G_1 is an additive group and G_2 is a multiplicative group. Let P be a generator of G_1. A bilinear pairing is a map $\hat{e} : G_1 \times G_1 \to G_2$ that satisfies the following properties [16]:

1. Bilinearity: $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$ for all $P, Q \in G_1$, $a, b \in \mathbb{Z}_p^*$.
2. Non-degeneracy: There are $P, Q \in G_1$ such that $\hat{e}(P, Q) \neq 1$, where 1 is the identity element of G_2.
3. Computability: $\hat{e}(P, Q)$ can be efficiently computed for all $P, Q \in G_1$.

The modified Weil pairing and Tate pairing provide admissible maps of this kind. For more details, please refer to [16]. The security of our scheme relies on the following hard problems.

Definition 1. Given groups G_1 and G_2 with the same prime order p, a generator P of G_1 and a bilinear map $\hat{e} : G_1 \times G_1 \to G_2$,

- The computational Diffie–Hellman (CDH) problem in G_1 is to compute abP given (P, aP, bP).
- The bilinear Diffie–Hellman (BDH) problem in $(G_1, G_2, \hat{e})$ is to compute $T = \hat{e}(P, P)^{abc}$ given (P, aP, bP, cP).
- The decisional bilinear Diffie–Hellman (DBDH) problem in $(G_1, G_2, \hat{e})$ is to decide whether $T = \hat{e}(P, P)^{abc}$ or not given (P, aP, bP, cP) and an element $T \in G_2$. If $T = \hat{e}(P, P)^{abc}$, we denote it by DBDH $(P, aP, bP, cP, T) = \top$. Otherwise, we denote it by DBDH $(P, aP, bP, cP, T) = \bot$.
- The gap bilinear Diffie–Hellman (GBDH) problem in $(G_1, G_2, \hat{e})$ is to compute $T = \hat{e}(P, P)^{abc}$ given (P, aP, bP, cP) with the help of the DBDH oracle that can decide if DBDH $(P, aP, bP, cP, T) = \top$ or $\bot$.
- The gap Diffie—Hellman (GDH) problem in G_1 is to compute abP given (P, aP, bP) with the help of the DBDH oracle that can decide if DBDH $(P, aP, bP, cP, T) = \top$ or $\bot$.

3. A heterogeneous signcryption scheme

In this section, we first describe the formal definition and security notions of heterogeneous signcryption scheme that allows a sender in the CLC environment to transmit a message to a receiver in the IBC environment. For simplicity, we use CI-HSC to denote this type of signcryption in the following content. Then we propose an efficient CI-HSC scheme and prove its security in the random oracle model.

3.1. Syntax

A generic CI-HSC scheme consists of the following eight algorithms.

Setup: It is a probabilistic algorithm performed by a PKG that takes as input a security parameter k, and outputs a master secret key s and the system parameters *params* that includes a master public key P_{pub}. For simplicity, we omit the system parameters *params* in the other algorithms in the following content.

CL-PPKE: It is a partial private key extraction algorithm run by the PKG that takes as input the master secret key s and a user's identity *ID*, and returns a partial private key D_{ID}. The PKG sends the partial private key to the user in a secure way.

CL-SVS: It is a secret value setup algorithm run by the users that takes as input an identity *ID*, and returns a secret value x_{ID}.

CL-PKS: It is a private key setup algorithm run by the users that takes as input a partial private key D_{ID} and a secret value x_{ID}, and outputs a full private key S_{ID}.

CL-PKG: It is a public key generation algorithm run by the users that takes as input a secret value x_{ID}, and outputs a public key PK_{ID}. The public key is published without a certificate.

IB-KE: It is a key extraction algorithm run by the PKG that takes as input the master secret key s and a user's identity *ID*, and returns a private key S_{ID}. The PKG sends the private key to the user in a secure way.

SC: It is a probabilistic signcryption algorithm performed by a sender that takes as input a plaintext message m, a sender's full private key S_{ID_s}, identity ID_s and public key PK_{ID_s} and a receiver's identity ID_r, and outputs a ciphertext σ.

USC: It is a deterministic unsigncryption algorithm performed by a receiver that takes as input a ciphertext σ, a sender's identity ID_s and public key PK_{ID_s} and a receiver's private key S_{ID_r} and identity ID_r, and outputs a plaintext m or a failure symbol $\perp$ if σ is not a valid ciphertext between the sender and the receiver.

The above algorithms should satisfy the consistency constraint of CI-HSC, i.e. if

$$\sigma = SC(m, S_{ID_s}, ID_s, PK_{ID_s}, ID_r),$$

then we have

$$m = USC(\sigma, ID_s, PK_{ID_s}, S_{ID_r}, ID_r).$$

Note that the *CL-PPKE*, *CL-SVS*, *CL-PKS* and *CL-PKG* are for users in the CLC environments and the *IBC-KE* algorithm is for users in the IBC environments. If the trusted third party in the CLC and IBC environments are different, we need to generate different master secret keys and corresponding master public key. In this paper, we think the trusted third party in the CLC and IBC environments is the same (please see Section 2). In such an assumption, the role of *CL-PPKE* and *IB-KE* is the same.

3.2. Security notions

A signcryption scheme should satisfy confidentiality (i.e. indistinguishability against adaptive chosen ciphertext attacks (IND-CCA2)) and unforgeability (i.e. existential unforgeability against adaptive chosen messages attacks (EUF-CMA)). We modify the notions in [24,25] slightly to adapt for CI-HSC.

For confidentiality, we consider the following game played between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$.

Initial: $\mathcal{C}$ runs *Setup* algorithm with a security parameter k and gives the system parameters *params* to $\mathcal{A}$.

Phase 1: $\mathcal{A}$ performs a polynomially bounded number of queries in an adaptive manner.

- Partial private key extraction queries: $\mathcal{A}$ chooses an identity *ID* and sends it to $\mathcal{C}$. $\mathcal{C}$ runs *CL-PPKE* algorithm and sends corresponding secret key D_{ID} to $\mathcal{A}$.
- Private key setup queries: When receiving an identity *ID* from $\mathcal{A}$, $\mathcal{C}$ runs *CL-PKS* algorithm and sends the full private key S_{ID} to $\mathcal{A}$ ($\mathcal{C}$ may first run *CL-PPKE* and *CL-SVS* algorithms if necessary).
- Public key queries: $\mathcal{A}$ chooses an identity *ID* and sends it to $\mathcal{C}$. $\mathcal{C}$ runs *CL-PKG* algorithm and sends the public key PK_{ID} to $\mathcal{A}$ ($\mathcal{C}$ may first run *CL-SVS* algorithm if necessary).
- Public key replacement queries: $\mathcal{A}$ may replace a public key PK_{ID} with a value chosen by it.
- Key extraction queries: When receiving an identity *ID* from $\mathcal{A}$, $\mathcal{C}$ runs *IB-KE* algorithm and sends the corresponding private key S_{ID} to $\mathcal{A}$.
- Signcryption queries: $\mathcal{A}$ chooses a message m, a sender's identity ID_i and a receiver's identity ID_j, $\mathcal{C}$ first runs *CL-PKS* and *CL-PKG* algorithms to get the private key S_{ID_i} and public key PK_{ID_i}, respectively. Then $\mathcal{C}$ sends the result of $SC(m, S_{ID_i}, ID_i, PK_{ID_i}, ID_j)$ to $\mathcal{A}$. It is possible that $\mathcal{C}$ does not know the sender's secret value, if the associated public key has been replaced. In this case, we require $\mathcal{A}$ to provide it.
- Unsigncryption queries: $\mathcal{A}$ chooses a ciphertext σ, a sender's identity ID_i and a receiver's identity ID_j, $\mathcal{C}$ first runs *IB-KE* and *CL-PKG* algorithms to get the private key S_{ID_j} and the public key PK_{ID_i}, respectively. Then $\mathcal{C}$ sends the result of $USC(\sigma, ID_i, PK_{ID_i}, S_{ID_j}, ID_j)$ to $\mathcal{A}$. The result is either a plaintext message m or a symbol $\perp$.

Challenge: $\mathcal{A}$ decides when phase 1 ends. $\mathcal{A}$ generates two equal length plaintexts (m_0, m_1), a sender's identity ID_s and a receiver's identity ID_r on which it wishes to be challenged. Note that ID_r should not be submitted to a key extraction query in the phase 1. $\mathcal{C}$ takes a random bit $\beta \in \{0, 1\}$ and computes $\sigma^* = SC(m_\beta, S_{ID_s}, ID_s, PK_{ID_s}, ID_r)$ which is sent to $\mathcal{A}$.

Phase 2: $\mathcal{A}$ may make a polynomially bounded number of queries adaptively again as in the phase 1. However, this time, it cannot make a key extraction query on ID_r and cannot make an unsigncryption query on (σ^*, ID_s, ID_r) to obtain the corresponding plaintext unless the public key PK_{ID_s} has been replaced after the challenge phase.

Guess: $\mathcal{A}$ produces a bit β' and wins the game if $\beta' = \beta$.

The advantage of $\mathcal{A}$ is defined as Adv $(\mathcal{A}) = |2\Pr[\beta' = \beta] - 1|$, where $\Pr[\beta' = \beta]$ denotes the probability that $\beta' = \beta$.

Definition 2. A CI-HSC scheme is $(\epsilon, t, q_{ppk}, q_{sk}, q_{pk}, q_{pkr}, q_k, q_s, q_u)$-IND-CCA2 secure if there does not exist a probabilistic t-polynomial time adversary $\mathcal{A}$ that has advantage at least ϵ after at most q_{ppk} partial private key extraction queries, q_{sk} private key setup queries, q_{pk} public key queries, q_{pkr} public key replacement queries, q_k key extraction queries, q_s signcryption queries and q_u unsigncryption queries in the confidentiality game.

The above definition grasps insider security for confidentiality of signcryption since the adversary knows all senders' private keys [26]. The insider security guarantees the forward security of signcryption scheme, i.e. confidentiality is kept in case the sender's private key is compromised.

For unforgeability, we need consider two types of adversaries, Type I and Type II [23,27], since the senders belong to the CLC en-

vironment. A Type I adversary models an attacker which is a common user and does not have the KGC's master secret key. But it can adaptively replace users' public keys with (valid) public keys of its choice. A Type II adversary models an honest-but-curious KGC who knows the KGC's master secret key. But it cannot replace users' public keys.

Now let us consider the unforgeability game played between a challenger $\mathcal{C}$ and the Type I adversary $\mathcal{F}_I$.

Initial: $\mathcal{C}$ runs *Setup* algorithm with a security parameter k and gives the system parameters *params* to $\mathcal{F}_I$.

Attack: $\mathcal{F}_I$ performs a polynomially bounded number of queries just like in the confidentiality game.

Forgery: $\mathcal{F}_I$ produces a ciphertext σ^*, a sender's identity ID_s and a receiver's identity ID_r and succeeds if the following conditions hold:

1. $USC(\sigma^*, ID_s, PK_{ID_s}, S_{ID_r}, ID_r) = m^*$.
2. $\mathcal{F}_I$ has not made a private key setup query for ID_s.
3. $\mathcal{F}_I$ cannot both make a public key replacement query for ID_s before forgery phase and make a partial private key extraction query in some phase.
4. $\mathcal{F}_I$ has not asked a signcryption query on (m^*, ID_s, ID_r).

The advantage of $\mathcal{F}_I$ is defined as the probability that it wins.

Definition 3. A CI-HSC scheme is $(\epsilon, t, q_{ppk}, q_{sk}, q_{pk}, q_{pkr}, q_k, q_s, q_u)$-Type-I-EUF-CMA secure if there does not exist a probabilistic t-polynomial time adversary $\mathcal{F}_I$ that has advantage at least ϵ after at most q_{ppk} partial private key extraction queries, q_{sk} private key setup queries, q_{pk} public key queries, q_{pkr} public key replacement queries, q_k key extraction queries, q_s signcryption queries and q_u unsigncryption queries in the Type I unforgeability game.

Finally, let us consider the unforgeability game played between a challenger $\mathcal{C}$ and the Type II adversary $\mathcal{F}_{II}$.

Initial: $\mathcal{C}$ runs *Setup* algorithm with a security parameter k and gives both the system parameters *params* and the master secret key s to $\mathcal{F}_{II}$.

Attack: $\mathcal{F}_{II}$ performs a polynomially bounded number of private key setup queries, public key queries and signcryption queries just like in the confidentiality game. Note that we do not need the partial private key extraction, key extraction and unsigncryption queries since $\mathcal{F}_{II}$ can do it by itself.

Forgery: $\mathcal{F}_{II}$ outputs a ciphertext σ^*, a sender's identity ID_s and a receiver's identity ID_r and succeeds if the following conditions hold:

1. $USC(\sigma^*, ID_s, PK_{ID_s}, S_{ID_r}, ID_r) = m^*$.
2. $\mathcal{F}_{II}$ has not made a private key setup query for ID_s.
3. $\mathcal{F}_{II}$ has not asked a signcryption query on (m^*, ID_s, ID_r).

The advantage of $\mathcal{F}_{II}$ is defined as the probability that it succeeds.

Definition 4. A CI-HSC scheme is $(\epsilon, t, q_{sk}, q_{pk}, q_s)$-Type-II-EUF-CMA secure if there does not exist a probabilistic t-polynomial time adversary $\mathcal{F}_{II}$ that has advantage at least ϵ after at most q_{sk} private key setup queries, q_{pk} public key queries and q_s signcryption queries in the Type II unforgeability game.

Definition 5. A CI-HSC scheme is EUF-CMA secure if it is both Type I EUF-CMA and Type II EUF-CMA secure.

In the above Definitions 3 and 4, we also allow the adversary to know the receiver's private key S_{ID_r}. That is, the definition also grasps the insider security for unforgeability of signcryption [26].

3.3. The proposed scheme

The proposed CI-HSC consists of the following eight algorithms.

Setup: Given a security parameter k, the PKG selects an additive group G_1 and a multiplicative G_2 of the same prime order p, a generator P of G_1, a bilinear map $\hat{e}: G_1 \times G_1 \to G_2$, and four secure hash functions H_1: $\{0,1\}^* \to G_1$, H_2: $\{0,1\}^* \to \{0,1\}^n$, H_3: $\{0,1\}^* \to G_1$ and H_4: $\{0,1\}^* \to G_1$. Here n is the number of bits of a message to be sent. The PKG randomly selects a master secret key $s \in \mathbb{Z}_p^*$ and computes the corresponding public key $P_{pub} = sP$. The PKG publishes the system parameters $\{G_1, G_2, p, \hat{e}, n, P, P_{pub}, H_1, H_2, H_3, H_4\}$ and keeps s secret.

CL-PPKE: A user submits its identity ID to the PKG. The PKG computes $Q_{ID} = H_1(ID)$ and sends the partial private key $D_{ID} = sQ_{ID}$ to the user.

CL-SVS: A user with identity ID chooses a random $x_{ID} \in \mathbb{Z}_p$ as the secret value.

CL-PKS: Given a partial private key D_{ID} and a secret value x_{ID}, this algorithm returns the full private key $S_{ID} = (x_{ID}, D_{ID})$.

CL-PKG: Given a secret value x_{ID}, this algorithm returns the public key $PK_{ID} = x_{ID}P$.

IB-KE: A user submits its identity ID to the PKG. The PKG computes $Q_{ID} = H_1(ID)$ and sends the private key $S_{ID} = sQ_{ID}$ to the user.

SC: Given a message m, a sender's full private key S_{ID_s}, identity ID_s and public key PK_{ID_s}, and a receiver's identity ID_r, this algorithm works as follows.

1. Choose $r \in \mathbb{Z}_p^*$ randomly.
2. Compute $U = rP$ and $T = \hat{e}(P_{pub}, Q_{ID_r})^r$.
3. Compute $h = H_2(U, T, ID_r)$.
4. Compute $C = m \oplus h$.
5. Compute $X = H_3(U, C, ID_s, PK_{ID_s})$.
6. Compute $Y = H_4(U, C, ID_s, PK_{ID_s})$.
7. Compute $V = D_{ID_s} + rX + x_{ID_s}Y$.
8. Output the ciphertext $\sigma = (U, C, V)$.

USC: Given a ciphertext $\sigma = (U, C, V)$, a sender's identity ID_s and public key PK_{ID_s}, and a receiver's private key S_{ID_r} and identity ID_r, this algorithm works as follows.

1. Compute $X = H_3(U, C, ID_s, PK_{ID_s})$.
2. Compute $Y = H_4(U, C, ID_s, PK_{ID_s})$.
3. Check if

   $$\hat{e}(P, V) = \hat{e}(P_{pub}, Q_{ID_s})\hat{e}(U, X)\hat{e}(PK_{ID_s}, Y)$$

   holds. If yes, perform the following step 4. Otherwise, reject this ciphertext and output the symbol $\perp$.
4. Compute $T = \hat{e}(U, S_{ID_r})$.
5. Compute $h = H_2(U, T, ID_r)$.
6. Recover the message $m = C \oplus h$.

3.4. Security

We prove that the CI-HSC satisfies the confidentiality and unforgeability by following Theorems 1 and 2, respectively.

Theorem 1. *In the random oracle model, if there exist an adversary $\mathcal{A}$ that has a non-negligible advantage ϵ against the IND-CCA2 security of the CI-HSC when running in a time t and executing q_{ppk} partial private key extraction queries, q_{sk} private key setup queries, q_{pk} public key queries, q_{pkr} public key replacement queries, q_k key extraction queries, q_s signcryption queries, q_u unsigncryption queries and q_{H_i} queries to hash oracles H_i $(i = 1, 2, 3, 4)$, then we can construct an algorithm $\mathcal{C}$ that can solve the GBDH problem with an advantage*

$$\epsilon' \geq \frac{\epsilon}{q_{H_1}}\left(1 - \frac{q_s(q_s + q_{H_3})}{2^k}\right)$$

in a time $t' \leq t + O(q_s + q_u)t_p$, where t_p is the cost for one pairing operation.

Proof. In this proof, we show how $\mathcal{C}$ uses $\mathcal{A}$ as a subroutine to solve a random instance (P, aP, bP, cP) of the GBDH problem.

Initial: $\mathcal{C}$ gives $\mathcal{A}$ the system parameters *params* with $P_{pub} = aP$. Note that $\mathcal{C}$ does not know the a. In this game, a simulates the secret key of the PKG.

Phase 1: $\mathcal{C}$ acts as $\mathcal{A}$'s challenger in the above IND-CCA2 game. $\mathcal{C}$ keeps four lists L_1, L_2, L_3 and L_4 to simulate the hash oracles H_1, H_2, H_3 and H_4, respectively. In addition, $\mathcal{C}$ maintains a list L_k that is initially empty to keep the public key information. We assume that H_1 queries are different and that $\mathcal{A}$ will ask $H_1(ID)$ before the identity ID is used in the other queries. In addition, we assume that the sender's identity is different to the receiver's identity by irreflexivity assumption [28]. $\mathcal{C}$ chooses a random number $\lambda \in \{1, 2, \ldots, q_{H_1}\}$ and answers $\mathcal{A}$'s queries as follows.

- H_1 queries: $\mathcal{A}$ chooses an identity and submits it to $\mathcal{C}$. At the λth H_1 query, $\mathcal{C}$ answers by $H_1(ID_\lambda) = bP$ and inserts $(ID_\lambda, \perp)$ into the list L_1. For ith H_1 query ($i \neq \lambda$), $\mathcal{C}$ randomly chooses $e_i \in \mathbb{Z}_p^*$, inserts (ID_i, e_i) into the list L_1 and answers $H_1(ID_i) = e_iP$.
- H_2 queries: When $\mathcal{A}$ asks a H_2 query on (U_i, T_i, ID_i), $\mathcal{C}$ does the following steps:
  1. If DBDH $(aP, bP, cP, T_i) = \top$, $\mathcal{C}$ returns T_i and stops.
  2. If the list L_2 contains entries $(U_i, \star, ID_i, h_i)$ such that DBDH $(aP, bP, U_i, T_i) = \top$, $\mathcal{C}$ returns h_i and updates the symbol $\star$ with T_i. Note that in this case $ID_i = ID_\lambda$.
  3. If $\mathcal{C}$ reaches this point of execution, $\mathcal{C}$ randomly chooses a value and returns it to $\mathcal{A}$. The query and the answer will be stored in the list. Note that $\mathcal{C}$ should maintain the consistency and avoid collision for these answers.
- H_3 queries: For a H_3 query on $(U_i, C_i, ID_i, PK_{ID_i})$, $\mathcal{C}$ first checks if the list L_3 contains a tuple $(U_i, C_i, ID_i, PK_{ID_i}, t_i, t_iP)$. If such a tuple is found, $\mathcal{C}$ returns t_iP to $\mathcal{A}$. Otherwise, $\mathcal{C}$ chooses a random $t \in \mathbb{Z}_p^*$, inserts the $(U_i, C_i, ID_i, PK_{ID_i}, t, tP)$ into the list L_3, and returns tP to $\mathcal{A}$.
- H_4 queries: For a H_4 query on $(U_i, C_i, ID_i, PK_{ID_i})$, $\mathcal{C}$ first checks if the list L_4 contains a tuple $(U_i, C_i, ID_i, PK_{ID_i}, w_i, w_iP)$. If such a tuple is found, $\mathcal{C}$ returns w_iP to $\mathcal{A}$. Otherwise, $\mathcal{C}$ chooses a random $w \in \mathbb{Z}_p^*$, inserts the $(U_i, C_i, ID_i, PK_{ID_i}, w, wP)$ into the list L_4, and returns wP to $\mathcal{A}$.
- Partial private key extraction queries: $\mathcal{A}$ can choose an identity ID_i and ask its partial private key. If $ID_i = ID_\lambda$, then $\mathcal{C}$ fails and stops. Otherwise the list L_1 should contain (ID_i, e_i) for some e_i (this shows $\mathcal{C}$ previously answered $H_1(ID_i) = e_iP$). $\mathcal{C}$ returns the partial private key $D_{ID_i} = e_iaP$.
- Private key setup queries: When $\mathcal{A}$ asks a private key setup query on an identity ID_i, if $ID_i = ID_\lambda$, $\mathcal{C}$ aborts. Otherwise, $\mathcal{C}$ runs H_1 oracle to obtain (ID_i, e_i). Then $\mathcal{C}$ searches L_k for the entry $(ID_i, PK_{ID_i}, x_{ID_i})$ ($\mathcal{C}$ generates a new key pair if this entry does not exist) and returns $S_{ID_i} = (x_{ID_i}, e_iaP)$.
- Public key queries: $\mathcal{A}$ chooses an identity ID and sends it to $\mathcal{C}$. If the list L_k contains a tuple $(ID_i, PK_{ID_i}, x_{ID_i})$, then $\mathcal{C}$ returns PK_{ID_i} to $\mathcal{A}$. Otherwise, $\mathcal{C}$ chooses a random $x_{ID_i} \in \mathbb{Z}_p^*$, computes $PK_{ID_i} = x_{ID_i}P$, inserts $(ID_i, PK_{ID_i}, x_{ID_i})$ into the list L_k and returns PK_{ID_i} to $\mathcal{A}$.
- Public key replacement queries: $\mathcal{A}$ may replace a public key PK_{ID} with a value chosen by it. For a public key replacement query for (ID_i, PK_{ID_i}), $\mathcal{C}$ updates the list L_k with tuple $(ID_i, PK_{ID_i}, \perp)$.
- Key extraction queries: When receiving an identity ID_i from $\mathcal{A}$, if $ID_i = ID_\lambda$, then $\mathcal{C}$ fails and stops. Otherwise the list L_1 should contain (ID_i, e_i) for some e_i. $\mathcal{C}$ returns the private key $S_{ID_i} = e_iaP$.
- Signcryption queries: When $\mathcal{A}$ chooses a message m, a sender's identity ID_i and a receiver's identity ID_j and makes a signcryption query, $\mathcal{C}$ proceeds as follows.
  1. If $ID_i \neq ID_\lambda$, $\mathcal{C}$ first runs private key setup oracle to get S_{ID_i} and public key oracle to get PK_{ID_i}. Then $\mathcal{C}$ can answer $\mathcal{A}$ by simply running $SC(m, S_{ID_i}, ID_i, PK_{ID_i}, ID_j)$.
  2. If $ID_i = ID_\lambda$, $\mathcal{C}$ chooses $u, v \in \mathbb{Z}_p^*$, sets $U = vaP$, and computes $T = \hat{e}(U, S_{ID_j})$ ($\mathcal{C}$ could obtain S_{ID_j} by running key extraction oracle). Then $\mathcal{C}$ sets $V = uaP + w_iPK_{ID_i}$ and defines the hash value $H_3(U, C, ID_i, PK_{ID_i})$ as $v^{-1}(uP - Q_{ID_i})$ ($\mathcal{C}$ could obtain PK_{ID_i} by running public key oracle and w_i from the list L_4). $\mathcal{C}$ fails if $H_3(U, C, ID_i, PK_{ID_i})$ is already defined but this case only happens with probability $(q_s + q_{H_3})/2^k$. $\mathcal{C}$ makes the H_2 query on (U, T, ID_j) to get h and computes $C = m \oplus h$. Finally, $\mathcal{C}$ returns $\sigma = (U, C, V)$ to $\mathcal{A}$
- Unsigncryption queries: $\mathcal{A}$ chooses a ciphertext $\sigma = (U, C, V)$, a sender's identity ID_i and a receiver's identity ID_j, $\mathcal{C}$ proceeds as follows.
  1. Compute $X = H_3(U, C, ID_i, PK_{ID_i})$ and $Y = H_4(U, C, ID_i, PK_{ID_i})$ and check if
     $$\hat{e}(P, V) = \hat{e}(P_{pub}, Q_{ID_i})\hat{e}(U, X)\hat{e}(PK_{ID_i}, Y)$$
     holds. If yes, perform the following step 2. Otherwise, reject this query and output the symbol $\perp$.
  2. If $ID_j \neq ID_\lambda$, $\mathcal{C}$ runs the key extraction oracle to get S_{ID_j} and computes $T = \hat{e}(U, S_{ID_j})$. $\mathcal{C}$ then runs H_2 oracle to get $h = H_2(U, T, ID_j)$ and returns $m = C \oplus h$.
  3. If $ID_j = ID_\lambda$, $\mathcal{C}$ cannot get the S_{ID_j} by the key extraction oracle. In this case, T cannot be computed. To return a consistent answer, $\mathcal{C}$ goes through the list L_2 and looks for a tuple (U, T, ID_j, h), for different values of T, such that DBDH $(aP, bP, U, T) = \top$. If such an entry exists, the correct T is found and returns $m = C \oplus h$.
  4. If $\mathcal{C}$ reaches this point of execution, it places the entry $(U, \star, ID_j, h)$ for a random h on the list L_2 and returns $m = C \oplus h$. The symbol $\star$ denotes an unknown value of T. Note that the identity component of all entries with the symbol $\star$ is ID_λ.

Challenge: $\mathcal{A}$ generates two equal length plaintexts (m_0, m_1), a sender's identity ID_s and a receiver's identity ID_r on which it wishes to be challenged. If $ID_r \neq ID_\lambda$, $\mathcal{C}$ aborts. Otherwise $\mathcal{C}$ takes a random bit $\beta \in \{0, 1\}$ and signcrypts m_β. To do so, it chooses a random hash value h^* and sets $U^* = cP$, $C^* = m_\beta \oplus h^*$ and $V^* = D_{ID_s} + rX + x_{ID_s}Y = D_{ID_s} + tcP + wPK_{ID_s}$, where D_{ID_s} can be obtained by running private key setup oracle, t is obtained from the list L_3, and w is obtained from the list L_4. $\mathcal{C}$ sends the challenge ciphertext $\sigma^* = (U^*, C^*, V^*)$ to $\mathcal{A}$.

Phase 2: $\mathcal{A}$ makes a polynomially bounded number of queries adaptively again as in the phase 1 with the limitation that: (1) it cannot make a key extraction query on ID_r; (2) it cannot ask an unsigncryption query on (σ^*, ID_s, ID_r) to obtain the corresponding plaintext unless the public key PK_{ID_s} has been replaced after the challenge phase. $\mathcal{C}$ answer $\mathcal{A}$'s queries as in the phase 1.

Guess: $\mathcal{A}$ produces a bit β' which is ignored by $\mathcal{C}$.

Since the list L_1 contains no more than q_{H_1} elements, $\mathcal{A}$ will output the identity ID_λ with probability $1/q_{H_1}$. If this event happens, the simulation is perfect unless $\mathcal{A}$ makes a H_2 query on the challenge-related tuple (U^*, T^*, ID_λ). Since the hash function H_2 is modeled as a random oracle, $\mathcal{A}$ will not have any advantage if this tuple does not appear on the list L_2. However, if this case happens, $\mathcal{C}$ will solve the GBDH problem due to the first step in the simulation of H_2. In the whole simulation, $\mathcal{C}$ makes at most $q_{H_2}^2 + q_{H_2}q_u$ DBDH oracle. □

Theorem 2. *In the random oracle model, the proposed scheme satisfies the EUF-CMA security under the CDH assumptions.*

Proof. This theorem follows from Lemmas 1 and 2. □

Lemma 1. *In the random oracle model, if there exist an adversary $\mathcal{F}_I$ that has a non-negligible advantage ϵ against the Type I EUF-CMA security of the CI-HSC when running in a time t and performing q_{ppk} partial private key extraction queries, q_{sk} private key setup queries,*

q_{pk} public key queries, q_{pkr} public key replacement queries, q_k key extraction queries, q_s signcryption queries, q_u unsigncryption queries and q_{H_i} queries to hash oracles H_i ($i = 1, 2, 3, 4$), then we can construct an algorithm $\mathcal{C}$ that can solve the GDH problem with an advantage

$$\epsilon' \geq \frac{\epsilon}{q_{H_1}}\left(1 - \frac{q_s(q_s + q_{H_3})}{2^k}\right)$$

in a time $t' \leq t + O(q_s + q_u)t_p$, where t_p is the cost for one pairing operation.

Proof. In this proof, we show how $\mathcal{C}$ uses $\mathcal{F}_I$ as a subroutine to solve a random instance (P, aP, bP) of the GDH problem.

Initial: $\mathcal{C}$ gives $\mathcal{F}_I$ the system parameters *params* with $P_{pub} = aP$. Note that $\mathcal{C}$ does not know the a. In this game, a simulates the secret key of the PKG.

Attack: $\mathcal{C}$ answers $\mathcal{F}_I$'s queries according to the proof of Theorem 1 except the H_2 queries. When $\mathcal{F}_I$ asks a H_2 query on (U_i, T_i, ID_i), $\mathcal{C}$ first checks if the list L_2 contains a tuple (U_i, T_i, ID_i, h_i). If such a tuple is found, $\mathcal{C}$ returns h_i to $\mathcal{F}_I$. Otherwise, $\mathcal{C}$ chooses a random $h \in \{0, 1\}^n$, inserts the (U_i, T_i, ID_i, h) into the list L_2, and returns h to $\mathcal{F}_I$.

Forgery: $\mathcal{F}_I$ outputs a ciphertext $\sigma^* = (U^*, C^*, V^*)$, a sender's identity ID_s and a receiver's identity ID_r. $\mathcal{C}$ checks if $ID_s = ID_\lambda$. If not, it aborts. Otherwise, it retrieves t from the list L_3 by querying H_3 on $(U^*, C^*, ID_s, PK_{ID_s})$ and w from the list L_4 by querying H_4 on $(U^*, C^*, ID_s, PK_{ID_s})$. Note that if $\mathcal{F}$ wins this game, the ciphertext σ^* must be valid. That is, we have

$$\begin{aligned}\hat{e}(P, V^*) &= \hat{e}(P_{pub}, Q_{ID_s})\hat{e}(U^*, tP)\hat{e}(PK_{ID_s}, wP)\\ &= \hat{e}(aP, bP)\hat{e}(U^*, tP)\hat{e}(PK_{ID_s}, wP)\end{aligned}$$

Thus $\mathcal{C}$ can compute

$$abP = V^* - tU^* - wPK_{ID_s}.$$

Let us now analyze the probability that $\mathcal{C}$ succeeds in solving the GDH problem instance. The probability that $\mathcal{C}$ aborts the simulation is related with the following events:

- E_1: $\mathcal{F}_I$ does not choose the identity ID_λ.
- E_2: $\mathcal{F}_I$ made a private key setup query on ID_λ.
- E_3 $\mathcal{F}_I$ made a public key replacement query for ID_s before the challenge phase and make a partial private key extraction query for ID_s in some phase.
- E_4: $\mathcal{C}$ aborts in a signcryption query because of a collision on H_3.

We know that $\Pr[\neg E_1] = 1/q_{H_1}$ and $\Pr[E_4] \leq q_s(q_s + q_{H_3})/2^k$. In addition, we know that $\neg E_1$ implies $\neg E_2$ and $\neg E_3$.

Therefore, we have

$$\epsilon' \geq \frac{\epsilon}{q_{H_1}}\left(1 - \frac{q_s(q_s + q_{H_3})}{2^k}\right).$$

In the whole simulation, $\mathcal{C}$ makes at most $q_{H_2}q_u$ DBDH oracle. □

Lemma 2. *In the random oracle model, if there exist an adversary $\mathcal{F}_{II}$ that has a non-negligible advantage ϵ against the Type II EUF-CMA security of the CI-HSC when running in a time t and executing q_{sk} private key setup queries, q_{pk} public key queries, q_s signcryption queries and q_{H_i} queries to hash oracles H_i ($i = 1, 2, 3, 4$), then we can construct an algorithm $\mathcal{C}$ that can solve the CDH problem with an advantage*

$$\epsilon' \geq \frac{\epsilon}{q_{sk} + q_{pk} + q_s + 1}\left(1 - \frac{q_s(q_s + q_{H_3})}{2^k}\right).$$

in a time $t' \leq t + O(q_s + q_u)t_p$, where t_p is the cost for one pairing operation.

Proof. We show how $\mathcal{C}$ uses $\mathcal{F}_{II}$ as a subroutine to solve a random instance (P, aP, bP) of the CDH problem.

Initial: $\mathcal{C}$ gives $\mathcal{F}_{II}$ the system parameters *params* with $P_{pub} = sP$. Here s is chosen randomly by $\mathcal{C}$.

Attack: $\mathcal{C}$ simulates $\mathcal{F}_{II}$'s challenger in the Type II EUF-CMA game. $\mathcal{C}$ keeps four lists L_1, L_2, L_3 and L_4 to simulate the hash oracles H_1, H_2, H_3 and H_4, respectively. In addition, $\mathcal{C}$ maintains a list L_k that is initially empty to keep the public key information. We assume that public key queries are different and that $\mathcal{F}_{II}$ will ask $H_1(ID)$ before the identity ID is used in the other queries. In addition, we assume that the sender's identity is different to the receiver's identity by irreflexivity assumption [28]. $\mathcal{C}$ chooses a random number $\lambda \in \{1, 2, \ldots, q_{sk} + q_{pk} + q_s + 1\}$. $\mathcal{C}$ answers H_2, H_3 and signcryption queries according to the same method of the Theorem 1. The other queries are explained as follows.

- H_1 queries: When $\mathcal{F}_{II}$ asks a H_1 query on ID_i, $\mathcal{C}$ first checks if the list L_1 contains a pair (ID_i, e_i). If such a pair is found, $\mathcal{C}$ returns e_iP to $\mathcal{F}_{II}$. Otherwise, $\mathcal{C}$ chooses a random $e \in \mathbb{Z}_p^*$, inserts the (ID_i, e) into the list L_1, and returns eP to $\mathcal{F}_{II}$.
- H_4 queries: For a H_4 query on $(U_i, C_i, ID_i, PK_{ID_i})$, $\mathcal{C}$ first checks if the list L_4 contains a tuple $(U_i, C_i, ID_i, PK_{ID_i}, w_i, w_ibP)$. If such a tuple is found, $\mathcal{C}$ returns w_ibP to $\mathcal{F}_{II}$. Otherwise, $\mathcal{C}$ chooses a random $w \in \mathbb{Z}_p^*$, inserts the $(U_i, C_i, ID_i, PK_{ID_i}, w, wbP)$ into the list L_4, and returns wbP to $\mathcal{F}_{II}$.
- Private key setup queries: When $\mathcal{F}_{II}$ asks a private key setup query on an identity ID_i, if $ID_i = ID_\lambda$, $\mathcal{C}$ aborts. Otherwise, $\mathcal{C}$ runs H_1 oracle to obtain (ID_i, e_i). Then $\mathcal{C}$ searches L_k for the entry $(ID_i, PK_{ID_i}, x_{ID_i})$ ($\mathcal{C}$ generates a new key pair if this entry does not exist) and returns $S_{ID_i} = (x_{ID_i}, se_iP)$.
- Public key queries: $\mathcal{F}_{II}$ chooses an identity ID_i and sends it to $\mathcal{C}$. If $ID_i \neq ID_\lambda$, $\mathcal{C}$ chooses a random $x_{ID_i} \in \mathbb{Z}_p^*$, computes $PK_{ID_i} = x_{ID_i}P$, inserts $(ID_i, PK_{ID_i}, x_{ID_i})$ into the list L_k and returns PK_{ID_i} to $\mathcal{F}_{II}$. Otherwise, $\mathcal{C}$ returns aP and inserts $(ID_\lambda, aP, \perp)$ into the list L_k.

Forgery: $\mathcal{F}_{II}$ outputs a ciphertext $\sigma^* = (U^*, C^*, V^*)$, a sender's identity ID_s and a receiver's identity ID_r. $\mathcal{C}$ checks if $ID_s = ID_\lambda$. If not, it aborts. Otherwise, it retrieves t from the list L_3 by querying H_3 on $(U^*, C^*, ID_s, PK_{ID_s})$ and w from the list L_4 by querying H_4 on $(U^*, C^*, ID_s, PK_{ID_s})$. Note that if $\mathcal{F}_{II}$ wins this game, the ciphertext σ^* must be valid. That is, we have

$$\begin{aligned}\hat{e}(P, V^*) &= \hat{e}(P_{pub}, Q_{ID_s})\hat{e}(U^*, tP)\hat{e}(PK_{ID_s}, wbP)\\ &= \hat{e}(sP, Q_{ID_s})\hat{e}(U^*, tP)\hat{e}(aP, wbP)\end{aligned}$$

Therefore, we have

$$\hat{e}(P, V^*) = \hat{e}(P, sQ_{ID_s})\hat{e}(tU^*, P)\hat{e}(P, wabP)$$

Thus $\mathcal{C}$ can compute

$$abP = w^{-1}(V^* - tU^* - sQ_{ID_s})$$

Let us now analyze the probability that $\mathcal{C}$ succeeds in solving the CDH problem instance. The probability that $\mathcal{C}$ aborts the simulation is related with the following events:

- E_1: $\mathcal{F}_{II}$ does not choose the identity ID_λ.
- E_2: $\mathcal{F}_{II}$ made a private key setup query on ID_λ.
- E_3: $\mathcal{C}$ aborts in a signcryption query because of a collision on H_3.

We know that $\Pr[\neg E_1] = 1/(q_{sk} + q_{pk} + q_s + 1)$ and $\Pr[E_3] \leq q_s(q_s + q_{H_3})/2^k$. In addition, we know that $\neg E_1$ implies $\neg E_2$. Note that the maximum length of the list L_k is $q_{sk} + q_{pk} + q_s + 1$.

Therefore, we have

$$\epsilon' \geq \frac{\epsilon}{q_{sk} + q_{pk} + q_s + 1}\left(1 - \frac{q_s(q_s + q_{H_3})}{2^k}\right).$$

□

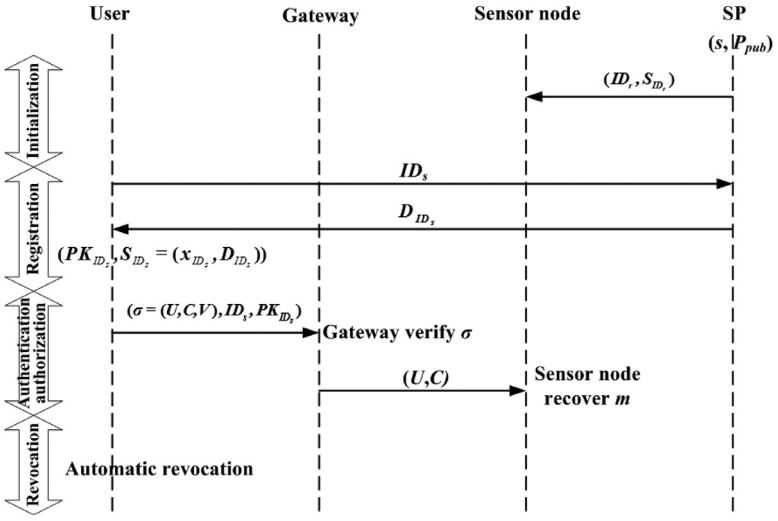

Fig. 2. The proposed access control scheme.

4. A practical access control scheme

In this section, we propose a practical access control scheme for the WSNs in the context of the IoT using the proposed CI-HSC scheme. Our scheme uses identity-based access control (IBAC) model [29,30]. The IBAC is a simple and practical access control model that associates access privilege with specific users. The proposed scheme consists of four phases: the initialization phase, the registration phase, the authentication and authorization phase, and the revocation phase. In this scheme, the SP plays the role of PKG in the IBC environment and the KGC in the CLC environment. The proposed access control scheme is summarized in Fig. 2.

4.1. Initialization phase

In this phase, the SP runs *Setup* algorithm and deploys the WSNs. Each sensor node is assigned an identity *ID* and a private key S_{ID} (the SP runs *IB-KE* algorithm to generate the private key S_{ID}).

4.2. Registration phase

An Internet user should register with the SP to obtain the access privilege of the WSNs. The user first submits its identity *ID* to the SP. Then the SP checks if the identity is valid. If the identity is not valid, the SP rejects this registration. Otherwise the SP runs *CL-PPKE* algorithm to get the partial private key $D_{ID} = sQ_{ID}$. After receiving D_{ID}, the Internet user runs *CL-SVS* algorithm to set the secret value $x_{ID} \in \mathbb{Z}_p^*$ and then runs *CL-PKS* to obtain the full private key $S_{ID} = (x_{ID}, D_{ID})$. Finally, the user runs *CL-PKG* algorithm to gets the public key $PK_{ID} = x_{ID}P$.

4.3. Authentication and authorization phase

We assume that an Internet user with identity ID_s hope to access the data of a sensor node with identity ID_r. The user first generates a query message m and runs the *SC* algorithm to get the ciphertext $\sigma = (U, C, V)$, where $U = rP$, $C = m \oplus h$, and $V = D_{ID_s} + rX + x_{ID_s}Y$. To resist the replay attack, we can concatenate the query message and a timestamp to form a new message that is signcrypted. Then the user sends the ciphertext, its identity ID_s and public key PK_{ID_s} to the gateway.

When receiving the query message from the user, the gateway first checks if

$$\hat{e}(P, V) = \hat{e}(P_{pub}, Q_{ID_s})\hat{e}(U, X)\hat{e}(PK_{ID_s}, Y)$$

holds. If the above equation does not hold, it refuses the query request. Otherwise, the user is authorized to access the data of the sensor node with identity ID_r. In this case, the gateway sends the (U, C) to the sensor node. The sensor node first computes $T = \hat{e}(U, S_{ID_r})$ and $h = H_2(U, T, ID_r)$. Then the sensor node recovers the message $m = C \oplus h$. Finally, the sensor node can encrypt the collected data using a symmetric cipher (such as AES [31]) with the session key h and send the data to the user. The session key h is only known by the user and the sensor node and assures the confidentiality for future communication between them. In this communication, confidentiality, integrity, authentication and non-repudiation are simultaneously achieved.

An important advantage of the proposed CI-HSC signcryption scheme is to achieves the public ciphertext authentication [32]. By using this signcryption scheme, the gateway can verify the validity and the origin of the ciphertext without knowing the message and getting any help from the intended receiver. Thus, we can move the most of computational cost of *USC* from the sensor nodes to the gateway. Of course, a weakness of our scheme is that the gateway may become the bottleneck. However, since the gateway is more powerful than the sensor nodes, our method is reasonable. If required, the anonymity also can be achieved by scrambling the user's identity ID_s and public key PK_{ID_s} together with the message at the fourth step of *SC* algorithm. That is, we compute $C = (ID_s||PK_{ID_s}||m) \oplus h$ instead of $C = m \oplus h$. Of course, we should modify the output value of H_2 to adapt the length of the encrypted message. Such changes do not affect the efficiency of our scheme.

4.4. Revocation

The registration can be revoked automatically by adding the expiration date in the identity. For example, when an Internet user first submits its identity *ID* to the SP. The SP uses a new identity "*ID*||2015-12-31" to generate the partial private key. That is, the SP computes $D_{ID} = sQ_{ID}$, where $Q_{ID} = H_1(ID||2015\text{-}12\text{-}31)$. Thus, the user only can access the WSNs before December 31, 2015. However, if we have to revoke a user's access privilege before the expiration date due to some reasons, the SP can send the revoked identity to the gateway. The gateway should keep a list of revoked identities to identify the validity of users.

4.5. Evaluation

In this section, we evaluate the performance and security of our scheme. First, we compare the computational cost and

Table 1
Comparison of performance.

Schemes	Computational cost			Sensor communication cost	
	User	Sensor	Gateway	Receive	Transmit
YHZXZ [13]	8M	3M	3M	$\|\mathbb{Z}_p^*\| + 2\|G_1\| + \|hash\| + 2\|ID\|$	$2\|\mathbb{Z}_p^*\| + \|G_1\| + \|hash\| + 3\|ID\|$
MXH [14]	2M	5M	–	$\|\mathbb{Z}_p^*\| + \|m\| + \|G_1\| + \|Cert\|$	–
Ours	3M + 1E + 1P	1P	4P	$\|m\| + \|G_1\|$	–

communication cost of our scheme with those of YHZXZ [13] and MXH [14] in Table 1.

We denote by M the point multiplication operation in G_1, E the exponentiation operation in G_2 and P the pairing operation. The other operations are ignored in Table 1 since these operations consume the most running time of the whole algorithm. $|x|$ denotes the number of bits of x. Because both YHZXZ and MXH are based on the PKI environment, we should verify the public key certificate before using a public key. Here we assume that the elliptic curve digital signature algorithm (ECDSA) [33] is used to sign a certificate. The ECDSA needs one point multiplication to sign a message and two point multiplications to verify a signature. Therefore, in YHZXZ, the gateway needs two point multiplications to verify the user's certificate and the user needs four point multiplications to verify the certificates of the gateway and sensor node. In MXH, the sensor node needs two point multiplications to verify the user's certificate. From Table 1, we know that our scheme has less computational cost than YHZXZ and more computational cost than MXH for the user. For the sensor node, our scheme has the least computational cost among the three scheme. For design an access control scheme for the WSNs in the context of the IoT, the most important issue is to reduce the computational cost of the sensor node since its resource is very limited. Therefore, our scheme is more practical than YHZXZ and MXH. For the communication cost of the sensor node, YHZXZ needs more cost since it is an interactive protocol. However, the sensor node does not need to receive the certificate of the user because the gateway assists to do it. In MXH, the sensor node should receive the user's certificate *Cert* to verify the validity. In our scheme, the sensor node does not need to receive the user's identity *ID* or certificate.

For both YHZXZ and MXH, we adopt the experiment result in [34] on MICA2 that is equipped with an ATmega128 8-bit processor clocked at 7.3728 MHz, 4 KB RAM and 128 KB ROM. From [34], we know that a point multiplication operation takes 0.81 s using an elliptic curve with 160 bits p that represents 80-bit security level. For our scheme, we adopt the experiment result in [35] on the same processor ATmega128. A pairing operation takes 1.9 s using the supersingular curve $y^2 + y = x^3 + x$ with an embedding degree 4 and implementing η_T pairing: $E(\mathbb{F}_{2^{271}}) \times E(\mathbb{F}_{2^{271}}) \rightarrow \mathbb{F}_{2^{4\cdot 271}}$, which is also equivalent to the 80-bit security level. According the results in [34,35], the computational time on the sensor node of YHZXZ, MXH and our scheme are $3 * 0.81 = 2.43$ s, $5 * 0.81 = 4.05$ s and $1 * 1.9 = 1.9$ s, respectively. As in [36,37], we assume that the power level of MICA2 is 3.0 V, the current draw in active mode is 8.0 mA, the current draw in receiving mode is 10 mA, the current draw in transmitting mode is 27 mA and the data rate is 12.4 kbps. For energy consumption, according to the method in [14,38], a point multiplication operation consumes $3.0 * 8.0 * 0.81 = 19.44$ mJ and a pairing operation consumes $3.0 * 8.0 * 1.9 = 45.6$ mJ. Therefore, the computational energy cost on the sensor node of YHZXZ, MXH and our scheme are $3 * 19.44 = 58.32$ mJ, $5 * 19.44 = 97.2$ mJ and $1 * 45.6 = 45.6$ mJ, respectively.

For the communication cost, we assume that $|m| = 160$ bits, $|hash| = 160$ bits and $|ID| = 80$ bits. In addition, the size of certificate is at least 688 bits [17]. For both YHZXZ and MXH, the size of an element in group G_1 is 1024 bits using an elliptic curve with 160 bits p. By standard compression technique [35,37], the size of an element in group G_1 can be reduced to 65 bytes. So, in YHZXZ, the sensor node should receive

$$|\mathbb{Z}_p^*| + 2|G_1| + |hash| + 2|ID| \text{ bits} = 20 + 2 * 65 + 20 + 2 * 10 \text{ bytes} = 190 \text{ bytes}$$

messages and transmit

$$2|\mathbb{Z}_p^*| + |G_1| + |hash| + 3|ID| \text{ bits} = 2 * 20 + 65 + 20 + 3 * 10 \text{ bytes} = 155 \text{ bytes}$$

messages. In MXH, the sensor node should receive

$$|\mathbb{Z}_p^*| + |m| + |G_1| + |Cert| \text{ bits} = 20 + 20 + 65 + 86 \text{ bytes} = 191 \text{ bytes}$$

messages. Our scheme uses a curve over the binary field $\mathbb{F}_{2^{271}}$. The size of an element in group G_1 is 542 bits. By standard compression technique, the size can be reduced to 34 bytes. So in our scheme, the sensor node needs to receive

$$|m| + |G_1| \text{ bits} = 20 + 34 \text{ bytes} = 54 \text{ bytes}$$

messages. From [37], we know the sensor node consumes $3 * 27 * 8/12400 = 0.052$ mJ and $3 * 10 * 8/12400 = 0.019$ mJ to transmit and receive one byte messages, respectively. Therefore, in YHZXZ, the sensor communication energy consumption is $0.052 * 155 + 0.019 * 190 = 11.67$ mJ. In MXH, the communication energy consumption is $0.019 * 191 = 3.63$ mJ. In our scheme, the communication energy consumption is $0.019 * 54 = 1.03$ mJ. The total energy consumption of the three schemes are $58.32 + 11.67 = 69.99$ mJ, $97.2 + 3.63 = 100.83$ mJ and $45.6 + 1.03 = 46.63$ mJ, respectively.

The computational time and total energy consumption on the sensor node are summarized in Figs. 3 and 4, respectively. From Fig. 3, we know that the computational cost of our scheme is reduced by about 22% and 53% compared to YHZXZ and MXH, respectively. From Fig. 4, we know that the energy consumption of

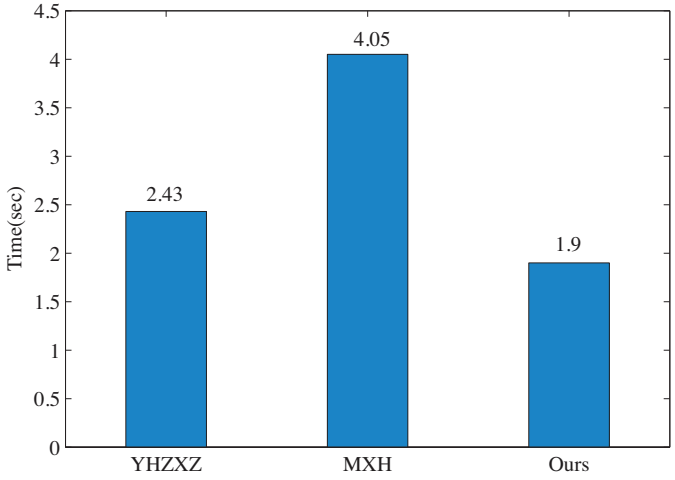

Fig. 3. The computational time of the sensor node.

Table 2
Comparison of security.

Schemes	Security							Environment
	Con	Int	Aut	Non	CipAut	InsSec	ForPro	
YHZXZ [13]	√	√	√	√	×	×	×	PKI → PKI
MXH [14]	√	√	√	√	√	×	√	PKI → PKI
Ours	√	√	√	√	√	√	√	CLC → IBC

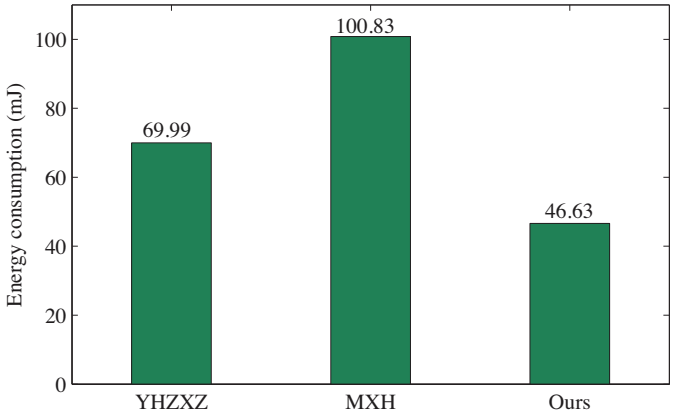

Fig. 4. The energy consumption of the sensor node.

our scheme is reduced by about 33% and 54% compared to YHZXZ and MXH, respectively. Of course, the computational cost of gateway in our scheme is higher than YHZXZ and MXH. We shift the computational cost of the sensor node to the gateway since our scheme has the ciphertext authenticity. The ciphertext authenticity allows the gateway to verify the ciphertext without the decryption.

We compare the security properties of the three schemes in Table 2. In the "Security" column, Con, Int, Aut, Non, CipAut, InsSec and ForPro denotes confidentiality, integrity, authentication, non-repudiation, ciphertext authenticity, insider security and formal proof, respectively. A symbol √ means that the scheme satisfies the security property and a symbol × means that the scheme does not satisfy the security property. Both YHZXZ and MXH do not satisfy the insider security [26] and our scheme has such security property. More importantly, our scheme allows a sender in the CLC environment to transmit a message to a receiver in the IBC environment, which conforms the characteristics of the WSNs in the context of the IoT.

5. Conclusions

In this paper, we proposed an efficient heterogeneous signcryption scheme that allows a sender in the CLC environment to transmit a message to a receiver in the IBC environment. We proved that the proposed scheme has the IND-CCA2 under the GBDH problem and EUF-CMA under the GDH and CDH problems in the random oracle model. In addition, we gave a practical access control scheme without certificates for the WSNs in the context of the IoT using the novel heterogeneous signcryption. As compared with existing YHZXZ and MXH using traditional signcryption, the computational cost of the sensor node in our scheme is reduced by about 22% and 53%, respectively and the energy consumption of the sensor node in our scheme is reduced by about 33% and 54%, respectively.

References

[1] C. Wang, C. Jiang, Y. Liu, X.Y. Li, S. Tang, Aggregation capacity of wireless sensor networks: Extended network case, IEEE Trans. Comput. 63 (6) (2014) 1351–1364.
[2] V.C. Gungor, G.P. Hancke, Industrial wireless sensor networks: Challenges, design principles, and technical approaches, IEEE Trans. Ind. Electr. 56 (10) (2009) 4258–4265.
[3] J. Niu, L. Cheng, Y. Gu, L. Shu, S.K. Das, R3e: Reliable reactive routing enhancement for wireless sensor networks, IEEE Trans. Ind. Inform. 10 (1) (2014) 784–794.
[4] R. Roman, J. Lopez, Integrating wireless sensor networks and the internet: A security analysis, Internet Res. 19 (2) (2009) 246–259.
[5] X.H. Le, S. Lee, I. Butun, M. Khalid, R. Sankar, M. Kim, M. Han, Y.K. Lee, H. Lee, An energy-efficient access control scheme forwireless sensor networks based on elliptic curve cryptography, J. Commun. Netw. 11 (6) (2009) 599–606.
[6] R.L. Rivest, A. Shamir, L. Adleman, A method for obtaining digital signatures and public-key cryptosystems, Commun. ACM 21 (2) (1978) 120–126.
[7] D. He, J. Bu, S. Zhu, S. Chan, C. Chen, Distributed access control with privacy support in wireless sensor networks, IEEE Trans. Wireless Commun. 10 (10) (2011) 3472–3481.
[8] R.L. Rivest, A. Shamir, Y. Tauman, How to leak a secret, Proceedings of Advances in Cryptology (ASIACRYPT'01), LNCS 2248, Springer-Verlag, 2001, pp. 552–565.
[9] J.K. Liu, M.H. Au, W. Susilo, J. Zhou, Linkable ring signature with unconditional anonymity, IEEE Trans. Knowl. Data Eng. 26 (1) (2014) 157–165.
[10] R. Zhang, Y. Zhang, K. Ren, Distributed privacy-preserving access control in sensor networks, IEEE Trans. Parallel Distrib. Syst. 23 (8) (2012) 1427–1438.
[11] S. Yu, K. Ren, W. Lou, FDAC: Toward fine-grained distributed data access control in wireless sensor networks, IEEE Trans. Parallel Distrib. Syst. 22 (4) (2011) 673–686.
[12] V. Goyal, O. Pandey, A. Sahai, B. Waters, Attribute-based encryption for fine–grained access control of encrypted data, in: Proceedings of ACM Conference on Computer and Communications Security (CCS'06), ACM, 2006, pp. 89–98.
[13] H. Yu, J. He, T. Zhang, P. Xiao, Y. Zhang, Enabling end-to-end secure communication between wireless sensor networks and the internet, World Wide Web 16 (4) (2013) 515–540.
[14] C. Ma, K. Xue, P. Hong, Distributed access control with adaptive privacy preserving property for wireless sensor networks, Secur. Commun. Netw. 7 (4) (2014) 759–773.
[15] Y. Zheng, Digital signcryption or how to achieve cost (signature & encryption) ≪ cost (signature) + cost(encryption), Proceedings of Advances in Cryptology (CRYPTO'97), LNCS 1294, Springer-Verlag, 1997, pp. 165–179.
[16] D. Boneh, M. Franklin, Identity-based encryption from the weil pairing, SIAM J. Comput. 32 (3) (2003) 586–615.
[17] K. Ren, W. Lou, K. Zeng, P.J. Moran, On broadcast authentication in wireless sensor networks, IEEE Trans. Wireless Commun. 6 (11) (2007) 4136–4144.
[18] D. He, C. Chen, S. Chan, J. Bu, SDRP: a secure and distributed reprogramming protocol for wireless sensor networks, IEEE Trans. Ind. Electr. 59 (11) (2012) 4155–4163.
[19] F. Li, D. Zhong, T. Takagi, Practical identity-based signature for wireless sensor networks, IEEE Wireless Commun. Lett. 1 (6) (2012) 637–640.
[20] H. Lu, J. Li, M. Guizani, Secure and efficient data transmission for cluster-based wireless sensor networks, IEEE Trans. Parallel Distrib. Syst. 25 (3) (2014) 750–761.
[21] F. Li, H. Zhang, T. Takagi, Efficient signcryption for heterogeneous systems, IEEE Syst. J. 7 (3) (2013) 420–429.
[22] F. Li, P. Xiong, Practical secure communication for integrating wireless sensor networks into the internet of things, IEEE Sensors J. 13 (10) (2013) 3677–3684.
[23] S.S. Al-Riyami, K.G. Paterson, Certificateless public key cryptography, Proceedings of Advances in Cryptology (ASIACRYPT'03), LNCS 2894, Springer-Verlag, 2003, pp. 452–474.
[24] P.S.L.M. Barreto, B. Libert, N. McCullagh, J.J. Quisquater, Efficient and provably-secure identity-based signatures and signcryption from bilinear maps, Advances in Cryptology (ASIACRYPT'05), LNCS 3788, Springer-Verlag, 2005, pp. 515–532.
[25] M. Barbosa, P. Farshim, Certificateless signcryption, in: Proceedings of ACM Symposium on Information, Computer and Communications Security (ASIACCS'08), ACM, 2008, pp. 369–372.
[26] J.H. An, Y. Dodis, T. Rabin, On the security of joint signature and encryption, Proceedings of Advances in Cryptology (EUROCRYPT'02), LNCS 2332, Springer-Verlag, 2002, pp. 83–107.

[27] F. Li, M. Shirase, T. Takagi, Certificateless hybrid signcryption, Math. Comput. Model. 57 (3-4) (2013) 324–343.
[28] X. Boyen, Multipurpose identity-based signcryption: a swiss army knife for identity-based cryptography, Proceedings of Advances in Cryptology (CRYPTO'03), LNCS 2729, Springer-Verlag, 2003, pp. 383–399.
[29] A. Herzberg, Y. Mass, J. Mihaeli, D. Naor, Y. Ravid, Access control meets public key infrastructure, or: assigning roles to strangers, in: Proceedings of IEEE Symposium on Security and Privacy (SP'00), 2000, pp. 2–14.
[30] V. Suhendra, A survey on access control deployment, Proceedings of Security Technology (SecTech'11), CCIS 259, Springer-Verlag, 2011, pp. 11–20.
[31] J. Daemen, V. Rijmen, The Design of Rijndael: AES-the Advanced Encryption Standard, Springer, 2002.
[32] S.S.M. Chow, S.M. Yiu, L.C.K. Hui, K.P. Chow, Efficient forward and provably secure ID-based signcryption scheme with public verifiability and public ciphertext authenticity, Proceedings of Information Security and Cryptology (ICISC'03), LNCS 2971, Springer-Verlag, 2004, pp. 352–369.
[33] D. Johnson, A. Menezes, S. Vanstone, The elliptic curve digital signature algorithm (ECDSA), Int. J. Inform. Secur. 1 (1) (2001) 36–63.
[34] N. Gura, A. Patel, A. Wander, H. Eberle, S.C. Shantz, Comparing elliptic curve cryptography and RSA on 8-bit CPUs, Proceedings of Cryptographic Hardware and Embedded Systems (CHES'04), LNCS 3156, Springer-Verlag, 2004, pp. 119–132.
[35] L.B. Oliveira, D.F. Aranha, C.P.L. Gouvêa, M. Scott, D.F. Câmara, J. López, R. Dahab, TinyPBC: Pairings for authenticated identity-based non-interactive key distribution in sensor networks, Comput. Commun. 34 (3) (2011) 485–493.
[36] X. Cao, W. Kou, L. Dang, B. Zhao, IMBAS: Identity-based multi-user broadcast authentication in wireless sensor networks, Comput. Commun. 31 (4) (2008) 659–667.
[37] K.A. Shim, Y.R. Lee, C.M. Park, EIBAS: An efficient identity-based broadcast authentication scheme in wireless sensor networks, Ad Hoc Netw. 11 (1) (2013) 182–189.
[38] K.A. Shim, s^2drp: Secure implementations of distributed reprogramming protocol for wireless sensor networks, Ad Hoc Netw. 19 (2014) 1–8.